LATENT SPACE STRUCTURING FOR CONDITIONAL TABULAR DATA GENERATION ON IMBALANCED DATASETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating synthetic tabular data under severe class imbalance is essential for domains where rare but high-impact events drive decision-making. Yet most generative models either overlook minority groups or fail to produce samples that are useful for downstream learning. We introduce CTTVAE, a Conditional Transformer-based Tabular Variational Autoencoder equipped with two complementary mechanisms: (i) a class-aware triplet margin loss that restructures the latent space for sharper intra-class compactness and inter-class separation, and (ii) a training-by-sampling strategy that adaptively increases exposure to underrepresented groups. Together, these components form CTTVAE+TBS, a framework that consistently yields more representative and utility-aligned samples without destabilizing training. Across six real-world benchmarks, CTTVAE+TBS achieves the strongest downstream utility on minority classes, often surpassing models trained on the original imbalanced data while maintaining competitive fidelity and privacy. Ablation studies further confirm that both latent structuring and targeted sampling contribute to these gains. By explicitly prioritizing downstream performance in rare categories, CTTVAE+TBS provides a robust and interpretable solution for conditional tabular data generation, with direct applicability to industries like healthcare, fraud detection, and predictive maintenance where even small gains on minority cases can be critical.

1 INTRODUCTION

Generating high-quality synthetic tabular data has become increasingly important for addressing challenges such as data scarcity, privacy constraints Borisov et al. (2022), and class imbalance. These issues are particularly critical in domains like healthcare Hernandez et al. (2022), fraud detection, and industrial monitoring, where rare but high-impact events, such as disease diagnosis, fraudulent transactions, or equipment failures, are severely underrepresented. Models trained on such imbalanced datasets often fail to capture meaningful minority-class patterns, leading to biased predictions and poor generalization D'souza et al. (2025). Given the ubiquity of tabular data, improving synthetic generation for downstream learning is a pressing need James et al. (2021).

Classical oversampling methods such as SMOTE Chawla et al. (2002) remain popular due to their simplicity, but they only interpolate between input-space samples and often yield unrealistic data in high dimensions Batista et al. (2004). Deep generative models (VAEs, GANs, and diffusion models) provide more expressive alternatives. Transformer-based VAEs Wang & Nguyen (2025) leverage self-attention to capture rich inter-feature dependencies, but they typically struggle with severe imbalance, producing poor-quality minority samples in low-density regions D'souza et al. (2025). Thus, two challenges remain: (i) generative models tend to overlook rare categories unless explicitly conditioned or regularized, and (ii) minority examples require latent representations that are both expressive and class-discriminative.

We propose the Conditional Transformer-based Tabular Variational Autoencoder (CTTVAE), a framework that combines latent space structuring with adaptive sampling to explicitly address class imbalance. CTTVAE incorporates a class-aware triplet margin loss to promote intra-class compactness and inter-class separation, and integrates a training-by-sampling (TBS) strategy that increases exposure to underrepresented groups, which will be referred as CTTVAE+TBS. Together, these mechanisms enable conditional generation that is both representative and utility-aligned,

particularly for minority categories. Unlike interpolation methods, CTTVAE operates in a structured latent space, producing semantically coherent samples without sacrificing training stability.

We evaluate CTTVAE across six public benchmarks, comparing it against two classical oversampling baselines and five state-of-the-art generative models. Our study provides a systematic analysis of fidelity, privacy, and downstream utility Alaa et al. (2022), and includes ablation experiments isolating the contributions of latent structuring and sampling. Results show that CTTVAE significantly improves downstream utility on minority classes while maintaining competitive fidelity and privacy.

The key contributions of this work are:

1. A conditional transformer-based VAE that explicitly improves minority-class utility through latent space structuring and targeted sampling.
2. Unlike prior models that either interpolate blindly in the input space or regularize the latent space without task awareness, CTTVAE explicitly restructures the latent manifold to reflect class semantics while simultaneously balancing exposure to rare groups.
3. A dual structuring that yields a controllable and general framework and extends naturally to any categorical conditioning variable, far beyond binary class imbalance.
4. Through extensive evaluation across six benchmarks, we demonstrate that CTTVAE consistently improves minority-class utility and privacy.

2 Related Work

2.1 Interpolation Methods

Traditional oversampling techniques serve as strong baselines for handling class imbalance. The Synthetic Minority Over-sampling Technique (SMOTE) Chawla et al. (2002) generates synthetic examples by linearly interpolating between minority class samples. Despite lacking the sophistication of deep models, this method can perform surprisingly well in combination with robust classifiers.

2.2 Deep Generative Models

Generative models for tabular data have emerged as powerful tools for addressing challenges such as data scarcity, privacy preservation, and class imbalance. Most high-performing models come from the 3 main generative model families: Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Diffusion models Kingma et al. (2013); Goodfellow et al. (2014); Ho et al. (2020). Among the early works in this area, CTGAN and TVAE Xu et al. (2019) introduced deep generative modeling frameworks specifically tailored to the tabular setting. CTGAN uses a conditional GAN architecture combined with mode-specific normalization to model mixed-type features and imbalanced class distributions, while TVAE formulates generation as a variational inference problem, enabling probabilistic modeling of heterogeneous feature types.

To improve the synthesis of mixed-type tabular data, CTAB-GAN Zhao et al. (2021) extends conditional GANs by introducing classification loss for better supervision, type-specific encoding for continuous and categorical variables, and lightweight preprocessing to handle long-tailed continuous distributions. Its design increases robustness to class imbalance and skewed data distributions. CopulaGAN, introduced in the SDV opensource library Patki et al. (2016), enhances CTGAN by combining it with a Gaussian copula-based normalization procedure.

Other recent methods such as Overlap Region Detection (ORD) D'souza et al. (2025) have shown that data imbalance often leads to poor generalization due to decision boundaries being dominated by majority-class instances. ORD addresses by selectively increasing the density of minority class data in critical regions of the data space, thereby improving classifier performance. Their results suggest that explicitly shaping the distribution of training samples can substantially enhance downstream utility, especially for underrepresented classes.

Recently, TabDDPM Kotelnikov et al. (2023) introduced diffusion-based generative modeling to the tabular domain, leveraging iterative denoising processes to achieve high-fidelity and privacy-aware samples. While TabDDPM reports state-of-the-art performance on several fidelity benchmarks, it does not support conditional generation by design.

Several other models have also been proposed for tabular data generation, including CTAB-GAN+ Zhao et al. (2024), TabSyn Zhang et al. (2023), MedGAN Choi et al. (2017), TabDiff Shi et al. (2025), and STaSy Kim et al. (2022), among others. All these methods highlight progress in realistic tabular generation, yet few tackle conditional synthesis under severe class imbalance.

3 METHODS

Our goal is to design a generative framework that explicitly improves the downstream utility of synthetic tabular data in imbalanced settings, with a particular focus on minority classes. To this end, we build on the TTVAE and introduce CTTVAE+TBS, which combines latent space structuring with adaptive sampling.

3.1 OVERVIEW OF TTVAE

TTVAE is a generative model for tabular data that extends the VAE framework by leveraging the Transformer's Vaswani et al. (2017) capabilities for heterogeneous tabular features Badaro et al. (2023). A Transformer-based encoder produces contextualized embeddings Huang et al. (2020), denoted $\mathbf{h}$, which capture both local and global dependencies between features. These embeddings allow the model to represent inter-feature relationships in a compressed format and seamlessly integrate categorical (one-hot encoded) and numerical (modeled through a Variational Gaussian Mixture) variables. Given an input $\mathbf{x}$, the encoder outputs:

$$\mathbf{h} = f_{\text{enc}}^{\text{Transf}}(\mathbf{x}), \quad \mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}), \tag{1}$$

where $\mathbf{h}$ captures inter-feature dependencies and $\mathbf{z}$ is sampled from the variational posterior. The decoder reconstructs $\mathbf{x}$ using both:

$$\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{h}). \tag{2}$$

Instead of the standard KL divergence term, TTVAE applies a Maximum Mean Discrepancy (MMD) penalty Gretton et al. (2012) between the aggregated posterior $q(\mathbf{z})$ and the Gaussian prior $p(\mathbf{z})$, yielding the objective:

$$\mathcal{L}_{\text{TTVAE}} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{h})] + \beta \cdot \text{MMD}(q(\mathbf{z}), p(\mathbf{z})), \tag{3}$$

where β controls the intensity of the MMD term. This formulation encourages a well-regularized latent space that captures higher-order moments and supports interpolation-based sampling. During generation, synthetic latent vectors are created via triangular interpolation over real latent encodings Fonseca & Bacao (2023), inspired by latent mixup Beckham et al. (2019), to promote semantic coherence and improve sample realism.

While TTVAE effectively models complex tabular structures, it lacks mechanisms to explicitly organize the latent space with respect to class information. As a result, it may struggle to generate useful samples for underrepresented classes when interpolation crosses ambiguous or low-density regions. This limitation motivates the need for class-aware latent structuring introduced in CTTVAE.

3.2 CTTVAE

As the first component of our proposed framework CTTVAE+TBS, CTTVAE extends TTVAE to structure the latent space with respect to class information. However, it is not inherently designed to prioritize or structure the latent space with respect to class or category-level semantics. This can limit their ability to generate useful samples for underrepresented groups, especially when generating data in ambiguous regions of the latent space. In comparison to ORD which operates in the data space, our approach takes a different perspective by directly structuring the latent space during training to encode class-aware relationships, enabling more reliable and controllable generation and improving sample quality of underrepresented classes.

To address this, we enhance the latent space geometry by incorporating triplet loss as it has proven to effectively work for VAEs Ishfaq et al. (2018), more specifically we implement the **triplet margin loss**. This addition encourages latent representations of instances from the same class to be embedded closely, while pushing apart samples from different classes. It directly acts on the mean latent vectors of the encoder.

Let $\mathbf{z}_a$ be the latent encoding of an anchor instance, $\mathbf{z}_p$ a positive sample from the same class, and $\mathbf{z}_n$ a negative sample from a different class. The triplet margin loss is defined as:

$$\mathcal{L}_{\text{triplet}} = \sum \max\left(\|\mathbf{z}_a - \mathbf{z}_p\|_2^2 - \|\mathbf{z}_a - \mathbf{z}_n\|_2^2 + m, 0\right) \tag{4}$$

where m is a margin hyperparameter. This objective encourages embeddings of the same class to lie closer together than those of different classes by at least margin m. We adopt **semi-hard negative mining** (full algorithm in appendix B), following Schroff et al. (2015), to guide the model towards informative comparisons, selecting $\mathbf{z}_n$ such that:

$$\|\mathbf{z}_a - \mathbf{z}_p\|_2^2 < \|\mathbf{z}_a - \mathbf{z}_n\|_2^2 < \|\mathbf{z}_a - \mathbf{z}_p\|_2^2 + m \tag{5}$$

The detailed procedure is summarized in Algorithm 1.

Algorithm 1 Semi-hard triplet mining procedure for CTTVAE

```
Compute pairwise distances: D ← cdist(μ, μ)
for i = 1 to n do
    a ← μ_i                                                        ▷ anchor
    label_a ← y_i
    PosIndices ← {j | y_j = y_i, j ≠ i}
    NegIndices ← {j | y_j ≠ y_i}
    if PosIndices = ∅ or NegIndices = ∅ then
        continue
    end if
    d_ap ← min{D[i][j] | j ∈ PosIndices}
    SemiHardMask ← {j ∈ NegIndices | d_ap < D[i][j] < d_ap + m}
    positive ← arg max{D[i][j] | j ∈ PosIndices}
    if SemiHardMask ≠ ∅ then
        negative ← random choice from SemiHardMask
    else
        negative ← arg min{D[i][j] | j ∈ NegIndices}
    end if
    Append triplet (a, positive, negative)
end for
Compute average triplet loss over valid triplets
```

The final training objective combines the TTVAE loss with the triplet margin loss:

$$\mathcal{L}_{\text{CTTVAE}} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{h})] + \beta \cdot \text{MMD}(q(\mathbf{z}), p(\mathbf{z})) + \alpha \cdot \mathcal{L}_{\text{triplet}}$$

where $\mathbf{x}$ is the input data, $\mathbf{h}$ is the contextual embedding produced by the Transformer encoder to capture inter-feature dependencies, and $\mathbf{z}$ is the latent representation sampled from the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$. The term $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{h})]$ represents the reconstruction loss. The term $\text{MMD}(q(\mathbf{z}), p(\mathbf{z}))$ represents the MMD loss. The hyperparameters β and α control the degree of intensity of the MMD term and the triplet loss term respectively.

This leads to a latent space that is better aligned with the desired class label eliminating the blending of unrelated samples. Furthermore, our framework allows the user to specify any categorical feature during training instead of class variable. This flexibility is especially valuable in use cases where the downstream task depends on factors other than the class label, such as demographic group, region, or product type.

Conditional Generation CTTVAE performs class-conditional generation by interpolating only within class-specific latent subsets (Figure 1). The encoder outputs (μ_i, σ_i, h_i) for each input x_i, and we then draw $z_i \sim \mathcal{N}(\mu_i, \text{diag}(\sigma_i^2))$.

For a target class c, we retain the subset $\mathcal{S}_c = \{(z_i, h_i) : y_i = c\}$. For each randomly chosen base $z_i \in \mathcal{S}_c$, we build a k-NN neighborhood $\mathcal{N}_k(z_i)$ (Minkowski metric; neighbors sorted by increasing distance). Denote the r-th neighbor by $\nu_{i,r}$. The *triangle* interpolation then draws a synthetic latent point via inverse-rank triangular weights with per-neighbor random scalars:

$$w_r = \frac{k-r}{\frac{k(k-1)}{2}} \quad (r = 1, \ldots, k), \qquad u_r \sim \mathcal{U}(0,1), \qquad \hat{z} = z_i + \sum_{r=1}^{k} w_r \, u_r \, (\nu_{i,r} - z_i). \tag{6}$$

The decoder receives both the synthetic latent vectors and the filtered encoder outputs and reconstructs $\hat{x} \sim p_\theta(x \mid \hat{z}, h)$. This ensures that generation remains confined to a coherent latent region aligned with the target class.

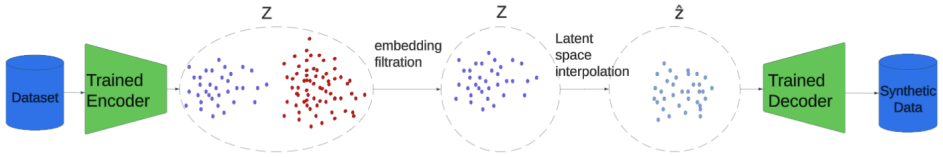

Figure 1: Conditional generation with CTTVAE. Encoder produces $\mathbf{h}$ and $\mathbf{z}$. Synthetic $\mathbf{z}$ are obtained by class-specific interpolation, then decoded with $\mathbf{h}$. Restricting interpolation to class regions preserves minority semantics and improves downstream utility.

This approach eliminates the need for a conditioning network, instead relying on the structurally aligned latent space learned during training. Since interpolation occurs within condition-specific regions, generated samples preserve class semantics and avoid blending across categories D'souza et al. (2025). While the conditioning mechanism is generalizable to any discrete feature, in this work we focus on the class label, as improving minority-class utility is our primary objective.

Our framework establishes a new paradigm in which the latent space is intentionally restructured for task relevance while the training process is guided to preserve minority representation. This coupling of geometric structuring and sampling control creates a generative framework explicitly tailored to imbalanced tabular learning, setting it apart from existing methods that either ignore class structure or rely on naive interpolation.

3.3 Training-by-Sampling (TBS)

Our second component, TBS, is a batch sampling strategy introduced in CTGAN Xu et al. (2019) to mitigate representation bias in tabular datasets, particularly when categorical features exhibit strong imbalance. Rather than drawing training batches uniformly at random, TBS constructs each batch by repeatedly selecting a specific value in a discrete column and sampling data points matching that value. This process ensures that all discrete values across all columns are regularly seen during training, even if their marginal frequency in the dataset is low.

We adopt a variant of the TBS concept, where sampling is guided solely by a user-specified categorical feature rather than sampling over all discrete columns. We do it on only the class label to address the imbalance to have a smoothed class sampling distribution. Specifically, we form a convex combination between the original class distribution P_{orig} and the uniform distribution P_{uniform}. The resulting sampling probability mass function (PMF) for each class c is given by:

$$\text{PMF}[c] = \lambda \cdot P_{\text{orig}}[c] + (1 - \lambda) \cdot P_{\text{uniform}}[c], \tag{7}$$

where $\lambda \in [0, 1]$ is a tunable hyperparameter. $\lambda = 1$ samples from the original class proportions, while $\lambda = 0$ does uniform sampling. Intermediate values offer a trade-off that improves exposure to rare classes without discarding the underlying data distribution, to mitigate risks of overfitting to the minority class.

4 Results

Datasets We extensively evaluate our methods against existing alternatives across various datasets with binary target variables, different properties, sizes, and number of features to evaluate models in different real-world settings, as seen in Table 1. The first three datasets have been used in most of the literature regarding tabular data generation, the other three have been chosen to explore more extreme cases.

Table 1: Summary of the datasets used in our experiments. CH = Churn Modeling, AD = Adult, DE = Default of Credit Card Clients, CR = Credit Card Fraud Detection (50k instances - due to limited resources, we undersampled the majority class while keeping the same number of minority instances), MA = Machine Predictive Maintenance, VE = Vehicle Insurance Claims. "IR" denotes the imbalance ratio between the majority and minority class in the training set.

Abbr.	Train/Test	#Num. Features	#Cat. Features	Target Column	IR
CH	8k / 2k	6	4	"Exited"	3.9
AD	24,111 / 6,028	6	8	"income"	3.0
DE	24k / 6k	20	3	"default.payment.next.month"	3.5
CR	40,378 / 10,095	29	0	"Class"	105.7
MA	8k / 2k	5	1	"Target"	28.5
VE	12,080 / 3,020	1	29	"FraudFound_P"	15.9

Experiments Each dataset is processed independently by each method to generate synthetic data that reflects the training distribution. The training and test subsets are split to have the same imbalance ratio as the full dataset. We did 25 hyperparameter tuning trials for all generative methods (see appendix G for more details), including the α and β hyperparameters from our loss function. We run our methods on A100 GPUs.

4.1 Utility Scores

To assess the utility of the synthetic data for downstream tasks, we employ the Machine Learning Efficacy (MLE) score. It evaluates the similarity in classification performance when models are trained on synthetic data and tested on real data, compared to models trained and tested entirely on real data. We compute the average F1 score using CatBoost Prokhorenkova et al. (2018), averaging results over three independent generations for each method and dataset. A higher MLE indicates better alignment with real-data performance, suggesting greater practical utility of the synthetic data.

Table 2 summarizes the results for minority classes. Across all datasets, our method achieves consistently the top 2 MLE scores on all datasets. In particular, it has the best result for 5 out of 6 datasets and the 2nd best for the remaining one. We outperform other generative models by significant margins, especially for highly imbalanced datasets. In comparison with TTVAE, our extension significantly improves performance on all datasets. These improvements are obtained while keeping majority-class performance stable, which is the intended behavior for oversampling in imbalanced regimes. SMOTE remains a strong baseline for utility and outperforms SOTA models in other papers however, it lacks the ability to scale to high-dimensional data, provide no privacy safeguards, and cannot handle flexible conditional generation(Table 4). It outperfo CTTVAE addresses all three, showing why deep generative models are essential in practice despite surface-level similarity in some scores.

Table 2: Average MLE and standard deviation over three generations computed with CatBoost across datasets for each class group (Majority, Minority). **Bold** represents the best results on each dataset and underlined represents the second best results for minority samples only on each dataset. The performances on the majority class remains stable for all the considered methods. "Real" represents the scores trained on the original dataset. Higher means better.

Method	CH	AD	DE	CR	MA	VE
Real	0.607±0.001	0.728±0.002	0.468±0.003	0.893±0.001	0.790±0.004	0.112±0.002
CTGAN	0.559±0.042	0.677±0.001	0.459±0.020	0.428±0.161	0.327±0.010	0.011±0.010
TVAE	0.502±0.015	0.609±0.003	0.397±0.006	0.838±0.020	0.189±0.006	0.001±0.001
CopulaGAN	0.560±0.010	0.569±0.004	0.474±0.038	0.450±0.190	0.302±0.023	0.053±0.060
CTABGAN	0.575±0.020	0.612±0.002	0.466±0.039	0.498±0.172	0.327±0.006	0.071±0.012
SMOTE	0.608±0.014	0.694±0.001	0.501±0.001	**0.891±0.001**	**0.678±0.035**	0.113±0.018
TTVAE	0.607±0.004	0.689±0.001	0.463±0.004	0.857±0.012	0.560±0.017	0.072±0.002
CTTVAE+TBS	**0.628±0.006**	**0.703±0.002**	**0.512±0.009**	0.881±0.004	**0.684±0.045**	**0.137±0.016**

4.2 Fidelity Analysis

We evaluate the fidelity of the synthetic data using three metrics: Wasserstein Distance (WD), Jensen–Shannon Divergence (JSD), and pairwise correlation error (see appendix A.2 for details).

Table 3 shows that interpolation methods yield on average the strongest fidelity scores overall. SMOTE achieves the lowest WD, JSD, and correlation error which is expected since interpolated samples remain very close to existing records. Among deep generative models, TTVAE obtains the lowest WD and JSD with CTTVAE+TBS close behind. In particular, CTTVAE+TBS ranks second-best or comparable on most fidelity metrics, while offering the minority-class utility gains absent from TTVAE. Correlation error further highlights this balance with CTTVAE+TBS achieving errors slightly lower than TTVAE (2.11% vs. 2.14%), and substantially lower than GAN-based methods (6–12%).

Table 3: Per-class average WD, JSD, and overall average pairwise correlation error (%). **Bold** and underline indicate best and second-best results respectively. Lower means better.

Method	WD ↓		JSD ↓		Corr. (%) ↓
	Maj.	Min.	Maj.	Min.	Avg.
CTGAN	0.103	0.128	0.084	0.092	11.48
TVAE	0.135	0.272	0.141	0.178	6.46
CopulaGAN	0.123	0.167	0.092	0.100	12.81
CTABGAN	0.159	0.205	0.076	0.078	6.22
SMOTE	**0.031**	**0.056**	**0.009**	**0.019**	**1.43**
TTVAE	0.057	0.111	0.028	0.044	2.14
CTTVAE+TBS	0.065	0.093	0.035	0.048	2.11

Figure 2 supports these findings: CTTVAE and TTVAE consistently display the lightest heatmaps, indicating minimal deviation from the true correlation structure. In contrast, TVAE, CTGAN and CTABGAN show heavier distortions, confirming their higher correlation errors. These findings show that CTTVAE provides a strong fidelity–utility trade-off, maintaining near-best fidelity among generative models while clearly outperforming them on minority utility.

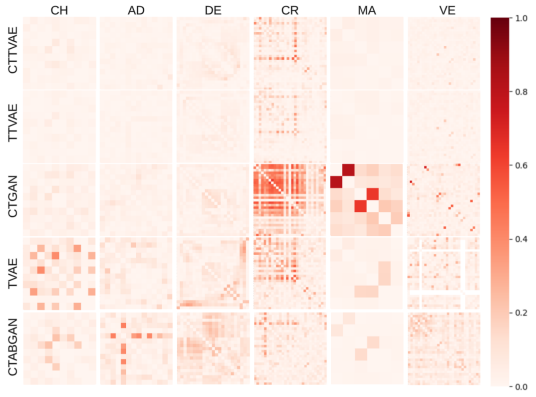

Figure 2: The absolute difference between correlation matrices computed on real and synthetic datasets. More intense red color indicates higher difference. Overall, CTTVAE and TTVAE capture correlations better.

4.3 Privacy Preservation

To evaluate potential privacy risks in the generated data, we rely on two Euclidean distance-based measures that focus on the proximity between synthetic and real samples. The Distance to Closest Record (DCR) quantifies the minimum distance from each synthetic record to its nearest real counterpart. Lower DCR values suggest a higher risk of memorization and worse privacy preservation. Complementing this, the Nearest Neighbour Distance Ratio (NNDR) assesses how distinct a synthetic

point is by comparing the distance to its 2 closest real neighbors. If the ratio is near one, the synthetic point is similarly distant from multiple real records, reducing the likelihood that it mimics any single example. We report the 5th percentile to follow the precedent established in prior work such as CTABGAN Zhao et al. (2021).

Table 4 compares CTTVAE+TBS against interpolation baselines, since interpolation directly biases these distance metrics. As expected, SMOTE exhibits the weakest privacy, nearly two times worse than the others, because convex combinations place synthetic points almost on top of real records. TTVAE has better privacy but CTTVAE+TBS achieves a clear margin with TTVAE and SMOTE for all classes and privacy metrics. With this we can deduce that latent-space restructuring combined with targeted sampling yields substantially stronger safeguards against memorization.

Full results against all other generative models are reported in Appendix Table 13. While some models report higher raw DCR values, this often reflects excessive drift away from real distributions, which correlates with poor utility and fidelity. By contrast, CTTVAE offers a balanced trade-off, maintaining strong privacy while clearly outperforming baselines on minority-class utility.

Table 4: Per-class Distance to Closest Record (DCR) and Nearest Neighbour Distance Ratio (NNDR) average across all datasets. **Bold** represents the best results and underline represents the second best on each metric. Higher values indicate better privacy.

Method	DCR ↑		NNDR ↑	
	Maj.	Min.	Maj.	Min.
SMOTE	0.380	0.864	0.282	0.372
TTVAE	0.699	1.382	0.368	0.440
CTTVAE+TBS	**1.587**	**1.511**	**0.534**	**0.543**

4.4 Ablation Study

We conduct an ablation study to disentangle the contributions of the triplet loss and the TBS strategy. Table 5 reports results relative to TTVAE across all datasets.

First, adding triplet loss (CTTVAE vs. TTVAE) yields consistent gains in minority utility (+0.032 on average) while maintaining stable performance on majority classes which shows that restructuring the latent space toward class separation produces more task-relevant minority samples. Importantly, CTTVAE also improves privacy with a much higher DCR/NNDR which reduces the risk of generating records overly close to real samples.

Second, incorporating TBS further amplifies these effects. CTTVAE+TBS achieves the largest overall gains on minority utility (+0.048), while keeping majority performance nearly unchanged. TBS also strengthens privacy across both majority and minority classes and, despite minor fluctuations, preserves fidelity at a competitive level. Figure 3 shows that while majority-class performance is stable across λ values, minority-class scores benefit substantially from balanced sampling, highlighting the importance of controlled exposure.

The results of the ablation study further demonstrate that triplet loss improves minority class alignment in the latent space, while TBS provides robust training dynamics, and that their combination produces the best trade-off between utility, fidelity, and privacy.

Table 5: Ablation study results relative to TTVAE across all datasets. Higher is better for MLE, DCR, NNDR; lower is better for WD, JSD. **Bold** represents the best result and underline represents the second best result.

Method	Avg. MLE ↑		Avg. WD ↓		Avg. JSD ↓		DCR ↑		NNDR ↑		Corr. (%) ↓
	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	
TTVAE+TBS	**0**	+0.030	+0.017	–0.009	+0.013	+0.006	+0.075	–0.012	+0.025	+0.035	+0.24
CTTVAE	-0.002	+0.032	**+0.005**	+0.010	+0.008	**–0.001**	**+0.888**	**+0.452**	+0.159	+0.087	+0.21
CTTVAE+TBS	-0.001	**+0.048**	+0.008	**–0.018**	**+0.007**	+0.004	**+0.888**	+0.129	**+0.166**	**+0.103**	**–0.03**

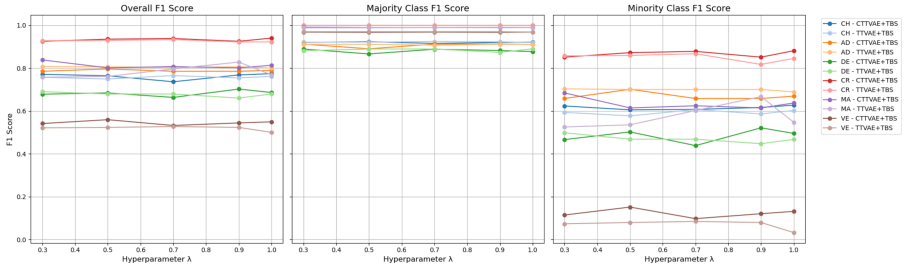

Figure 3: Impact on the minority class of the sampling hyperparameter λ on F1 scores across datasets for CTTVAE+TBS and TTVAE+TBS. $\lambda = 1$ represents the models performances without aplying TBS. Performance on minority classes depends greatly on its value.

5 Limitations and Discussion

Our framework demonstrates consistent utility improvements across all datasets with strong gains for minority classes showing that structuring the latent space with triplet loss and balancing exposure through TBS are effective strategies for generating task-relevant data under imbalance. Importantly, these benefits come without degrading majority-class performance, which makes the method particularly suitable for domains where minority events drive downstream decisions.

Some trade-offs remain. The triplet loss introduces computational overhead, which may limit scalability to very large datasets unless more efficient mining strategies are adopted. Fidelity metrics also show that class-aware interpolation can underperform raw TTVAE in distributional alignment, while privacy scores indicate that interpolation-based models inherently place synthetic samples closer to real points. However, these effects are moderate, and the addition of TBS mitigates them by reducing overfitting and improving privacy without destabilizing training. Crucially, in imbalanced learning scenarios, slightly lower fidelity is an acceptable compromise when it yields substantially higher utility and stronger privacy protection, since the practical value of synthetic data lies in improving downstream task performance while avoiding direct memorization. We argue this trade-off is not a drawback since this is more valuable for downstream deployment, where the goal is robust minority-class decision making rather than pixel-perfect distribution matching.

We included GAN and VAE-based models since they can be reproduced efficiently under our per-class protocol. In contrast, diffusion models such as TabDDPM require orders of magnitude more resources to rerun and are only reported at dataset level, making them impractical to align with our evaluation.

6 Conclusion

We introduced CTTVAE, a conditional transformer-based VAE that establishes a new paradigm for imbalanced tabular data generation by restructuring the latent space and guiding training to preserve minority representation. This structuring and adaptive sampling yields consistent improvements in downstream utility for rare classes while also enhancing privacy and keeping fidelity competitive. Unlike interpolation baselines that appear strong only because they produce samples close to real records, CTTVAE+TBS achieves a more meaningful balance, generating diverse, task-relevant data. These properties make it a practical solution for real-world domains such as fraud detection, predictive maintenance, and healthcare, where minority utility and privacy protection are paramount.

7 Future Work

While this study confirms the effectiveness of structuring latent spaces and sampling bias, several avenues remain open. TBS enhances performances but requires to tune its hyperparameter λ. A natural extension of this work involves exploring more self-adaptive sampling strategies to optimize class exposure dynamically based on training dynamics or dataset properties, reducing manual intervention while preserving performance gains. Extending the privacy evaluation with metrics such as Membership Inference Attack Accuracy would be beneficial as most papers don't use them.

REFERENCES

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pp. 290–306. PMLR, 2022.

Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 11:227–249, 2023.

Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

Christopher Beckham, Sina Honari, Vikas Verma, Alex M Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, and Chris Pal. On adversarial mixup resynthesis. *Advances in neural information processing systems*, 32, 2019.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pp. 286–305. PMLR, 2017.

Annie D'souza, M Swetha, and Sunita Sarawagi. Synthetic tabular data generation for imbalanced classification: The surprising effectiveness of an overlap class. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16127–16134, 2025.

Joao Fonseca and Fernando Bacao. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10(1):115, 2023.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

Haque Ishfaq, Assaf Hoogi, and Daniel Rubin. Tvae: Triplet-based variational autoencoder using metric learning. *arXiv preprint arXiv:1802.04403*, 2018.

Stefanie James, Chris Harbron, Janice Branson, and Mimmi Sundler. Synthetic data use: exploring use cases to optimise data utility. *Discover Artificial Intelligence*, 1(1):15, 2021.

Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. *arXiv preprint arXiv:2210.04018*, 2022.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pp. 17564–17579. PMLR, 2023.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pp. 399–410. IEEE, 2016.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a mixed-type diffusion model for tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex X Wang and Binh P Nguyen. Ttvae: Transformer-based generative modeling for tabular data generation. *Artificial Intelligence*, pp. 104292, 2025.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. *arXiv preprint arXiv:2310.09656*, 2023.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pp. 97–112. PMLR, 2021.

Zilong Zhao, Aditya Kunar, Robert Birke, Hiek Van der Scheer, and Lydia Y Chen. Ctab-gan+: Enhancing tabular data synthesis. *Frontiers in big Data*, 6:1296508, 2024.

A Experimental Setup

A.1 Machine Learning Efficacy Models

For the Machine Learning Efficacy (MLE) score, we conducted a more in-depth experimentation with several other traditional classifiers. We selected the following diverse set of 7 machine learning models (results are shown in appendix C):

RandomForest was implemented using the `RandomForestClassifier` from the `scikit-learn` library.

XGBoost was implemented using the `XGBClassifier` from the `xgboost` library.

LightGBM was implemented using the `LGBMClassifier` from the `lightgbm` library.

CatBoost was implemented using the `CatBoostClassifier` from the `catboost` library.

Logistic Regression was implemented using the `LogisticRegression` class from the `scikit-learn` library.

Support Vector Machines (SVM) was implemented using the `SVC` class from the `scikit-learn` library.

Multi-Layer Perceptrons (MLP) was implemented using the `MLPClassifier` class from the `scikit-learn` library.

A.2 Fidelity Metrics

- **Wasserstein Distance (WD):** quantifies the cost of transforming the real distribution into the synthetic one and is particularly sensitive to shifts in tails and distribution spread. Lower WD indicates more accurate modeling of class-conditional distributions.
- **Jensen-Shannon Divergence (JSD):** measures the dissimilarity between probability distributions in a symmetric and bounded way. It captures how well the synthetic data approximates the global support and entropy of the real distribution.
- **Pairwise Correlation Error:** evaluates the structural consistency of synthetic data by computing the absolute difference between real and synthetic Pearson correlation matrices. This metric reflects how well inter-feature relationships are preserved.

A.3 Privacy Metrics

Before computing privacy metrics (DCR and NNDR), we subsample 15% of real and synthetic data and apply z-score normalization. This ensures meaningful distance computations and consistency across datasets.

A.4 Pipeline

Figure 4 illustrates the experimental pipeline used in our study. The process begins with multiple tabular datasets, which are first preprocessed to ensure compatibility with all data generation models and downstream classifiers. This includes encoding categorical features, scaling numerical ones, and applying a fixed train/test split that preserves the original class imbalance ratio (IR).

The training set is then passed to a selected data generation methods. Each dataset is processed independently by each method to generate synthetic data that reflects the training distribution.

The synthetic data is then evaluated along 3 parallel axes: utility, fidelity and privacy analysis.

This dissected evaluation allows us to analyze each method's capacity to generate useful, faithful, and privacy-preserving synthetic data. The results are then aggregated and analyzed to draw conclusions about performance trade-offs and the effect of different techniques.

To ensure fair comparison, we fixed the random seed for all model initializations, training, and data splits. Each experiment was repeated with the same configuration across all methods.

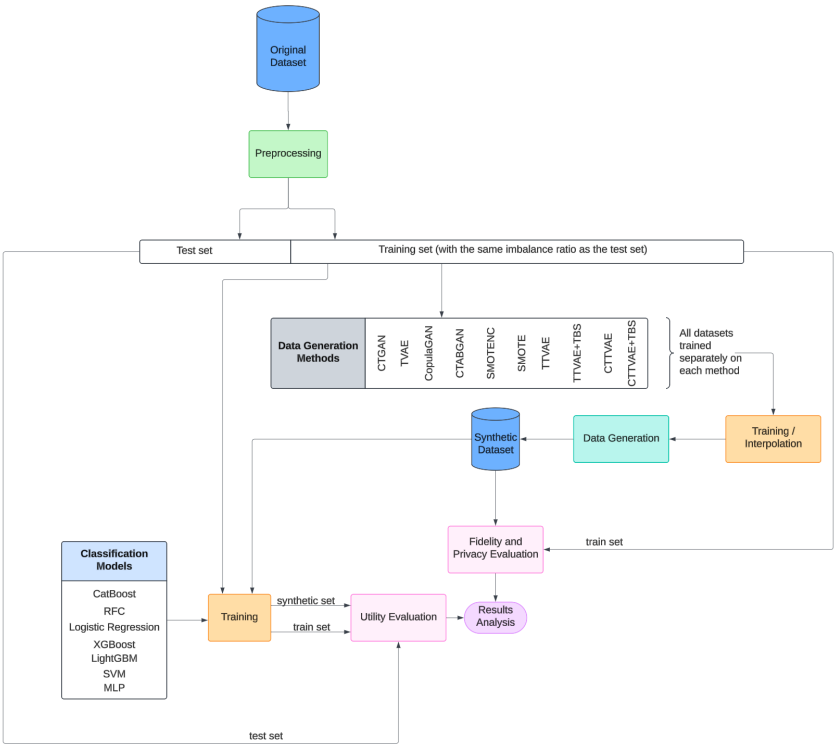

Figure 4: Pipeline

A.5 Implementation of Baseline Data Generation Methods

To evaluate the performance of our proposed method, we implemented several baseline data generation methods commonly used for synthetic tabular data generation. We describe the implementation details for each method:

SMOTE was implemented using the `SMOTE` class from the `imblearn` library. A customized function was implemented to generate an entirely synthetic dataset.

CTGAN was implemented using the `CTGANSynthesizer` class from the `sdv` library.

TVAE was implemented using the `TVAESynthesizer` class from the `sdv` library.

CopulaGAN was implemented using the `CopulaGANSynthesizer` class from the `sdv` library.

CTABGAN was implemented using the code from its repository and adapted to our pipeline.

TTVAE was implemented using the code from its repository and adapted to our pipeline.

B Dataset Preprocessing Details

All datasets used in this study are publicly available, and the corresponding preprocessing code is provided in the official repository, with dedicated notebooks for each dataset. Preprocessing involved only minimal cleaning: removing rows with missing values or duplicates, digitizing target columns, and dropping irrelevant features such as IDs.

C Additional Results

In Table 6, we compute the average F1 score across 7 classifiers for each method and dataset. A higher MLE indicates better alignment with real-data performance, suggesting greater practical utility of the synthetic data.

Table 6: The values of the average MLE and standard deviation for each method and each dataset averaged over all classifiers. Each classifier has been tuned and then trained 10 times (training not seeded) with the best set of hyperparameters on the same generated data. **Bold** represents the best results on each dataset and underlined represents the second best results on each dataset. "Real" represents the scores of the models trained on the original dataset. Higher means better.

Method	CH	AD	DE	CR	MA	VE
Real	0.732±0.001	0.811±0.001	0.670±0.001	0.945±0.001	0.826±0.004	0.530±0.002
CTGAN	0.697±0.001	0.785±0.001	0.676±0.001	0.815±0.004	0.633±0.002	0.489±0.002
TVAE	0.698±0.001	0.747±0.001	0.646±0.001	0.858±0.004	0.580±0.003	0.485±0.000
CopulaGAN	0.706±0.001	0.725±0.001	0.646±0.002	0.597±0.009	0.612±0.002	0.487±0.002
CTABGAN	0.711±0.002	0.752±0.001	0.667±0.001	0.836±0.004	0.606±0.006	0.518±0.002
SMOTE	0.735±0.001	0.797±0.001	**0.685±0.001**	**0.943±0.001**	**0.799±0.003**	**0.537±0.002**
TTVAE	0.735±0.001	0.797±0.001	0.656±0.002	0.925±0.001	0.710±0.006	0.507±0.002
CTTVAE+TBS	**0.744±0.001**	**0.801±0.001**	**0.683±0.001**	0.927±0.004	0.774±0.004	0.531±0.002

Table 7 reports the mean and standard deviation of the MLE scores over three generations, separated by majority and minority classes. As expected, majority-class performance remains highly stable across all methods, with very low variance, while minority-class results show larger fluctuations reflecting the higher sensitivity to imbalance. This confirms that our improvements primarily benefit the minority class without degrading performance on the majority.

Table 7: Average MLE and standard deviation computed with CatBoost across datasets for each class group (Majority, Minority). **Bold** represents the best results on each dataset and underlined represents the second best results for minority samples only on each dataset. Its performance on the majority class remains stable for all the considered methods. "Real" represents the scores of CatBoost trained on the original dataset. Higher means better.

(a) Majority Class Only

Method	CH	AD	DE	CR	MA	VE
Real	0.923±0.001	0.918±0.002	0.890±0.003	0.999±0.001	0.994±0.004	0.970±0.002
CTGAN	0.887±0.004	0.906±0.007	0.870±0.003	0.993±0.002	0.949±0.004	0.970±0.001
TVAE	0.889±0.001	0.881±0.005	0.885±0.002	0.998±0.001	0.982±0.001	0.970±0.001
CopulaGAN	0.856±0.001	0.894±0.004	0.883±0.003	0.981±0.002	0.940±0.003	0.970±0.001
CTABGAN	0.894±0.002	0.896±0.004	0.868±0.004	0.998±0.001	0.983±0.002	0.962±0.002
SMOTE	0.917±0.001	0.908±0.001	0.882±0.001	0.999±0.001	0.990±0.001	0.969±0.001
TTVAE	0.919±0.001	0.910±0.003	0.890±0.002	0.999±0.001	0.989±0.002	0.968±0.001
CTTVAE+TBS	0.920±0.001	0.910±0.002	0.882±0.002	0.999±0.001	0.991±0.001	0.967±0.001

(b) Minority Class Only

Method	CH	AD	DE	CR	MA	VE
Real	0.607±0.001	0.728±0.002	0.468±0.003	0.893±0.001	0.790±0.004	0.112±0.002
CTGAN	0.559±0.042	0.677±0.001	0.459±0.020	0.428±0.161	0.327±0.010	0.011±0.010
TVAE	0.502±0.015	0.609±0.003	0.397±0.006	0.838±0.020	0.189±0.006	0.001±0.001
CopulaGAN	0.560±0.010	0.569±0.004	0.474±0.038	0.450±0.190	0.302±0.023	0.053±0.060
CTABGAN	0.575±0.020	0.612±0.002	0.466±0.039	0.498±0.172	0.327±0.006	0.071±0.012
SMOTE	0.608±0.014	0.694±0.001	0.501±0.001	**0.891±0.001**	**0.678±0.035**	0.113±0.018
TTVAE	0.607±0.004	0.689±0.001	0.463±0.004	0.857±0.012	0.560±0.017	0.072±0.002
CTTVAE+TBS	**0.628±0.006**	**0.703±0.002**	**0.512±0.009**	0.881±0.004	**0.684±0.045**	**0.137±0.016**

The per-class Wasserstein Distance results across datasets are presented in Table 8, separated into moderately and highly imbalanced datasets.

	CH		**AD**		**DE**	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	0.109	0.125	0.131	0.110	0.074	0.082
TVAE	0.242	0.268	0.184	0.204	0.107	0.331
CopulaGAN	0.142	0.148	0.136	0.177	0.103	0.162
CTABGAN	0.187	0.182	0.312	0.337	0.200	0.243
TTVAE	**0.032**	0.064	0.067	0.097	0.066	0.103
TTVAE+TBS	0.041	0.060	0.074	0.091	0.101	0.108
CTTVAE	0.044	<u>0.046</u>	0.063	0.096	0.085	0.106
CTTVAE+TBS	0.039	0.056	<u>0.052</u>	0.072	0.089	0.116
SMOTE	<u>0.037</u>	**0.040**	**0.051**	**0.065**	**0.042**	<u>0.063</u>

(a) Moderately imbalanced datasets

	CR		**MA**		**VE**	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	0.136	<u>0.186</u>	0.080	0.150	0.090	0.106
TVAE	<u>0.101</u>	0.260	0.125	0.210	0.053	0.362
CopulaGAN	0.181	0.256	0.105	0.176	0.071	0.083
CTABGAN	0.128	0.231	0.035	0.133	0.096	0.105
TTVAE	0.143	0.249	**0.013**	0.099	0.023	0.052
TTVAE+TBS	0.186	0.200	0.017	<u>0.072</u>	0.029	0.047
CTTVAE	0.137	0.210	<u>0.013</u>	0.158	0.065	0.078
CTTVAE+TBS	0.133	0.147	0.015	0.091	0.065	0.074
SMOTE	**0.019**	**0.078**	0.019	**0.048**	<u>0.020</u>	<u>0.040</u>

(b) Highly imbalanced datasets

Table 8: Wasserstein Distance per class averaged over three generations across datasets. **Bold** represents the best results and <u>underline</u> represents the second best. Lower is better.

The per-class Jensen-Shannon Divergence scores across datasets are shown in Table 9. The CR dataset is omitted from this table due to its lack of categorical features.

	CH		AD		DE	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	0.024	0.025	0.101	0.104	0.116	0.086
TVAE	0.224	0.232	0.089	0.097	0.157	0.172
CopulaGAN	0.028	0.033	0.104	0.107	0.103	0.094
CTABGAN	0.052	0.057	0.143	0.140	0.057	0.070
TTVAE	0.012	0.019	0.039	0.051	0.040	0.035
TTVAE+TBS	0.016	0.022	0.068	0.066	0.064	0.083
CTTVAE	0.009	0.018	0.041	0.056	0.073	0.067
CTTVAE+TBS	0.009	0.016	0.045	0.058	0.067	0.060
SMOTE	**0.004**	**0.012**	**0.009**	0.018	**0.003**	0.008

(a) Moderately imbalanced datasets

	MA		VE	
	Maj.	Min.	Maj.	Min.
CTGAN	0.066	0.092	0.115	0.151
TVAE	0.114	0.091	0.118	0.296
CopulaGAN	0.100	0.123	0.124	0.144
CTABGAN	0.030	**0.008**	0.098	0.115
TTVAE	0.017	0.060	0.031	0.055
TTVAE+TBS	0.023	0.035	0.031	0.042
CTTVAE	0.025	0.022	0.053	0.070
CTTVAE+TBS	0.018	0.050	0.038	0.060
SMOTE	0.010	0.011	**0.020**	0.044

(b) Highly imbalanced datasets

Table 9: Jensen-Shannon Divergence per class averaged over three generations across datasets. **Bold** represents the best results and underline represents the second best. Lower scores are better. CR dataset is omitted since it does not contain categorical features.

Table 10 reports the pairwise correlation error rates across datasets. SMOTE achieves the lowest correlation errors in most cases, particularly on CR and DE. CTTVAE and its TBS variant also perform well, with notably low errors and comparable with the baseline interpolation methods. In contrast, models like CTGAN and CopulaGAN show higher deviation from the real data's correlation structure.

Method	CH	AD	DE	CR	MA	VE
CTGAN	2.89	2.40	3.39	25.71	29.95	4.55
TVAE	8.97	4.86	5.21	11.29	3.93	4.48
CopulaGAN	3.15	3.32	5.85	32.32	27.90	4.37
CTABGAN	3.94	6.89	8.54	6.79	3.19	7.99
TTVAE	1.12	**1.05**	2.31	5.20	1.39	1.76
TTVAE+TBS	1.43	1.14	3.22	5.31	1.43	1.52
CTTVAE	1.08	1.37	2.31	6.31	1.14	1.67
CTTVAE+TBS	1.13	1.40	2.29	5.23	0.95	1.63
SMOTE	**1.05**	1.18	**1.78**	**1.82**	1.32	**1.42**

Table 10: Pair-wise correlation error rate (%) averaged over three generations for each method across datasets. **Bold** represents the best results and underline represents the second best on each dataset. Lower scores means better.

Tables 11 12 report per-class privacy scores. Table 13 summarizes the results. Higher values indicate greater dissimilarity between synthetic and real records, which typically suggests better privacy preservation. However, high DCR and NNDR can sometimes reflect low data utility and fidelity if the synthetic samples drift too far from the true data distribution. For instance, COPULAGAN and CTABGAN achieve consistently among the highest scores but often performs poorly in terms

of utility. This does not imply that the synthetic data is of high quality. On the contrary, it instead signals poor alignment with the original data.

Among the generative models, TTVAE and CTTVAE variants tend to strike a more balanced profile, achieving moderate scores without overstepping into unrealistic territory given their high utility scores. In highly imbalanced settings, TTVAE-based model achieve strong comparable privacy scores w.r.t. other methods, suggesting that these methods and training strategies are more suitable for these types of datasets. Still, it is crucial to interpret DCR and NNDR jointly with fidelity and utility metrics as it does not paint the full picture.

	CH		**AD**		**DE**	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	0.692	0.993	<u>1.000</u>	0.912	0.712	0.876
TVAE	**1.501**	**1.664**	0.774	0.890	**1.009**	**1.778**
CopulaGAN	0.750	0.802	0.970	<u>1.055</u>	0.775	1.196
CTABGAN	<u>0.980</u>	<u>1.079</u>	**1.474**	**1.745**	<u>0.810</u>	<u>1.271</u>
TTVAE	0.179	0.344	0.390	0.483	0.359	0.656
TTVAE+TBS	0.194	0.565	0.469	0.556	0.542	0.841
CTTVAE	0.375	0.482	0.423	0.604	0.509	0.787
CTTVAE+TBS	0.380	0.480	0.468	0.628	0.650	0.489
SMOTE	0.356	0.517	0.368	0.482	0.302	0.497

(a) DCR – Moderately imbalanced datasets

	CR		**MA**		**VE**	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	<u>2.613</u>	2.320	<u>0.505</u>	0.512	<u>8.753</u>	**9.910**
TVAE	1.799	**4.481**	0.447	<u>0.849</u>	7.244	0.750
CopulaGAN	2.189	2.042	**0.683**	**0.850**	**8.950**	<u>9.628</u>
CTABGAN	1.789	2.799	0.316	0.521	8.011	8.736
TTVAE	1.955	<u>3.160</u>	0.154	0.568	1.158	3.084
TTVAE+TBS	**2.622**	3.044	0.143	0.502	0.676	2.066
CTTVAE	1.488	2.569	0.240	0.590	6.489	5.974
CTTVAE+TBS	1.753	2.281	0.219	0.497	6.051	4.691
SMOTE	0.454	1.458	0.233	0.534	0.565	1.693

(b) DCR – Highly imbalanced datasets

Table 11: Average per-class privacy scores (DCR) over three generations across moderately and highly imbalanced datasets. **Bold** represents the best results and <u>underline</u> represents the second best on each dataset. Higher scores means better.

	CH		AD		DE	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	0.496	0.541	0.413	0.469	0.684	0.665
TVAE	**0.842**	**0.848**	<u>0.524</u>	<u>0.535</u>	**0.816**	**0.859**
CopulaGAN	0.540	0.511	0.490	0.479	0.709	0.699
CTABGAN	<u>0.612</u>	<u>0.588</u>	**0.606**	**0.748**	<u>0.761</u>	<u>0.748</u>
TTVAE	0.136	0.240	0.276	0.345	0.502	0.538
TTVAE+TBS	0.169	0.336	0.349	0.447	0.636	0.601
CTTVAE	0.342	0.327	0.340	0.402	0.618	0.582
CTTVAE+TBS	0.336	0.344	0.354	0.420	0.629	0.621
SMOTE	0.276	0.307	0.203	0.263	0.341	0.347

(a) NNDR – Moderately imbalanced datasets

	CR		MA		VE	
	Maj.	Min.	Maj.	Min.	Maj.	Min.
CTGAN	<u>0.877</u>	<u>0.893</u>	0.644	<u>0.685</u>	**0.878**	<u>0.888</u>
TVAE	0.807	0.834	**0.719**	0.567	0.851	0.540
CopulaGAN	**0.886**	**0.912**	<u>0.679</u>	**0.738**	**0.878**	**0.892**
CTABGAN	0.834	0.822	0.565	0.351	<u>0.866</u>	0.835
TTVAE	0.798	0.738	0.319	0.477	0.175	0.302
TTVAE+TBS	0.835	0.766	0.282	0.514	0.089	0.189
CTTVAE	0.756	0.790	0.404	0.481	0.715	0.584
CTTVAE+TBS	0.784	0.730	0.428	0.570	0.674	0.572
SMOTE	0.353	0.537	0.448	0.594	0.070	0.183

(b) NNDR – Highly imbalanced datasets

Table 12: Average NNDR per-class privacy scores over three generations across moderately and highly imbalanced datasets. **Bold** represents the best results and <u>underline</u> represents the second best on each dataset. Higher scores means better.

Method	DCR ↑		NNDR ↑	
	Maj.	Min.	Maj.	Min.
CTGAN	<u>2.379</u>	2.587	0.665	0.690
TVAE	2.129	1.933	**0.759**	**0.729**
CopulaGAN	**2.386**	<u>2.596</u>	0.697	<u>0.705</u>
CTABGAN	2.231	**2.691**	<u>0.708</u>	0.682
TTVAE	0.687	1.274	0.360	0.456
TTVAE+TBS	0.735	1.216	0.404	0.469
CTTVAE	1.269	1.456	0.510	0.492
CTTVAE+TBS	1.584	1.779	0.516	0.538
SMOTE	0.380	0.864	0.282	0.372
SMOTENC	0.860	1.132	0.388	0.429

Table 13: Per-class Distance to Closest Record (DCR) and Nearest Neighbour Distance Ratio (NNDR). **Bold** represents the best results and <u>underline</u> represents the second best on each dataset. Higher values indicate better privacy.

D Additional Visualizations

The heatmaps in Figure 5 provide a complementary view of fidelity by visualizing how well the correlation structure of the real data is preserved across models in the ablation study. As shown, CTTVAE+TBS maintains lighter patterns compared to alternatives, indicating lower deviation from the real correlation structure. This visualization confirms the quantitative fidelity results, where our proposed method remains competitive with the strongest baselines while offering superior utility for minority classes.

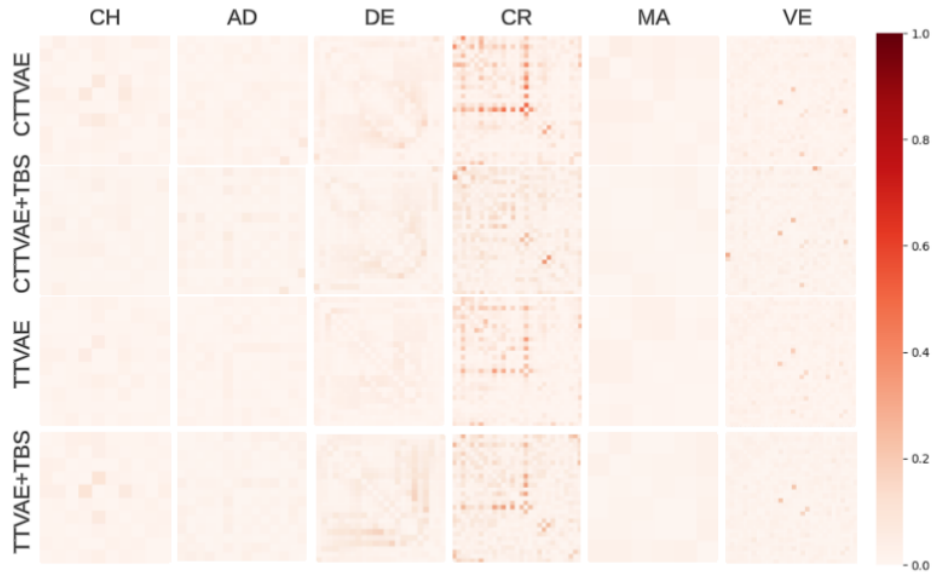

Figure 5: The absolute difference between correlation matrices computed on real and synthetic datasets for the ablation study. More intense red color indicates higher difference.

PCA projections in Figure 6 reveal that CTTVAE yields clearer class boundaries and tighter clusters than TTVAE, confirming the role of triplet loss in enabling coherent, class-aware generation under imbalance. Furthermore, we see that the clusters keep a non spherical shape, allowing for outliers to remain as such (as opposed to how contrastive losses separate the space). Maintaining outliers is important, especially in imbalanced settings since often those are the most important instances.

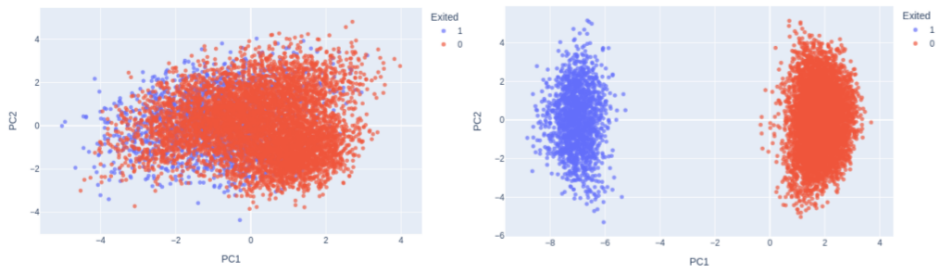

Figure 6: Latent space encoded by TTVAE (left) and CTTVAE (right) for the CH dataset, projected on a 2D space using PCA for visualization purposes.

E RUNTIME

We report the training and sampling time of CTTVAE and TTVAE for the Churn Modeling (CH) dataset for comparison (Table 14). Both models have been trained on A100 GPU. Although CTTVAE training is slower due to triplet mining, the overhead remains modest relative to modern GPU capabilities, and the resulting gains in minority-class utility outweigh this cost. Efficient mining or adaptive margins can further reduce runtime.

Model	batch_size	epochs	train_steps	training time
TTVAE	128	125	7,812	381s
CTTVAE	128	169	10,562	1,086s

(a) **Training time**

Model	number to sample	sample_time
TTVAE	8k	0.29s
CTTVAE	8k	0.33s

(b) **Sampling time**

Table 14: Training and sampling time for CTTVAE and TTVAE for the CH dataset.

F HYPERPARAMETER SEARCH SPACES

We performed hyperparameter optimization using Optuna library for both the downstream MLE classifiers (Table 15) and the generative models (Table 16). For each generative model, we conducted 25 trials to identify the best-performing configuration based on utility scores. Due to computational constraints, hyperparameter tuning for CTTVAE and TTVAE was divided into two stages: we first selected the best model configuration and then conducted a focused search on the L2 regularization scale for both models and the triplet loss factor for CTTVAE. For experiments involving TBS, we did not run a full 25-trial search; instead, we evaluated different values of the sampling hyperparameter λ using the previously selected best configuration for each model.

Model	Search Space
RandomForest	num estimators: Int[50, 300] max depth: Int[3, 20] min samples_split: Int[2, 10] min samples_leaf: Int[1, 10]
XGBoost	n_estimators: Int[50, 300] max_depth: Int[3, 20] learning rate: Float[0.01, 0.3]
LightGBM	num estimators: Int[50, 300] num leaves: Int[20, 100] learning rate: Float[0.01, 0.3]
CatBoost	iterations: Int[50, 300] depth: Int[3, 10] learning rate: Float[0.01, 0.3]
LogisticRegression	C: Float[0.01, 10.0] penalty: {l1, l2} solver: {liblinear, saga}
SVM	C: Float[0.01, 10.0] kernel: {linear, rbf}
MLP	hidden layer: {(100,), (50,50), (100,50)} activation: {relu, tanh} alpha: Float[1e-5, 1e-1] max iter: 500 (fixed)
Number of tuning trials	30

Table 15: Hyperparameter search space for classifier models used for MLE

Model	**Search Space**
CTGAN / CopulaGAN	pac: {1, 5, 10} batch_size: {64, 128, 256, 500} epochs: {50, 100, 150}
TVAE	batch_size: {64, 128, 256, 512} epochs: {10, 50, 100, 150}
CTABGAN	batch_size: {64, 128, 256} epochs: {150, 200, 250} class_dim: {128, 256} l2scale: Float[1e-6, 1e-3] learning rate: Float[1e-4, 1e-2] num_channels: {32, 64, 128} random_dim: {64, 100, 128}
TTVAE	batch_size: {16, 32, 64} epochs: {10, 50, 100, 150} latent_dim: {16, 32, 64} embedding_dim: {64, 128, 256} nhead: derived from (64,4), (128,4/8), (256,8) dim_feedforward: {512, 1024, 2048} dropout: Float[0.0, 0.3] l2scale: {1e-5, 1e-4, 1e-3}
CTTVAE	batch_size: {16, 32, 64} epochs: {10, 50, 100, 150} latent_dim: {16, 32, 64} embedding_dim: {64, 128, 256} nhead: derived from (64,4), (128,4/8), (256,8) dim_feedforward: {512, 1024, 2048} dropout: Float[0.0, 0.3] triplet_margin: Float[0.1, 1.0] l2scale: {1e-5, 1e-4, 1e-3} triplet_factor: {0.5, 1, 2, 5}
TBS	λ: {0.3, 0.5,0.7, 0.9}
Number of tuning trials	25

Table 16: Hyperparameter search space for deep generative models.